Санкт-Петербургский политехнический университет Петра Великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

**ОТЧЁТ ПО НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

Синтез, выбор и анализ значимости признаков в задаче Facial Anti-Spoofing Detection

Выполнил

Н. С. Щебнев

Руководитель

доцент, к.т.н.

Научный консультант

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 202\_\_ г.

Санкт-Петербург

2020

Содержание

[ВВЕДЕНИЕ 3](#_Toc59571672)

[1 Постановка задачи Facial Anti-Spoofing Detection 4](#_Toc59571673)

[2 Набор данных 6](#_Toc59571674)

[3 Синтез признаков 8](#_Toc59571675)

[4 Отбор статистически значимых признаков 12](#_Toc59571676)

[4.1 Однофакторный дисперсионный анализ ANOVA 13](#_Toc59571677)

[4.2 Bootstrap 15](#_Toc59571678)

[5 Анализ значимости признаков 16](#_Toc59571679)

[ЗАКЛЮЧЕНИЕ 24](#_Toc59571680)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc59571681)

[Приложение А 26](#_Toc59571682)

[Приложение Б 27](#_Toc59571683)

[Приложение В 28](#_Toc59571684)

# ВВЕДЕНИЕ

Биометрическая идентификация человека на сегодняшний день является очень популярным направлением, актуальность которого строится на использовании при распознавании человека его биологических уникальных особенностей: отпечатков пальцев, голос, рисунок сосудов сетчатки глаз, походка и лицо. Тем не менее, для систем аутентификации всегда остро стоит вопрос о безопасности, поэтому одной из главных задач этой области является детектирование атак злоумышленников. В задаче аутентификации пользователей по лицу одним из наиболее сложных видов атак является атака воспроизведением видео, на котором изображен пользователь.

В данной работе рассмотрены вопросы синтеза и выбора признаков изображения для более эффективной работы алгоритмов обнаружения атак.

# 1 Постановка задачи Facial Anti-Spoofing Detection

Задача Facial Anti-Spoofing Detection (Обнаружение не подменённых лиц) ставит перед собой целью классифицировать входное изображение на изображения, объектом сцены которых являются настоящие человеческие лица, и изображения, объектом сцены которых являются физические (распечатанные фотографии, маски) или электронные (экран мобильных устройств) носители, содержащие информацию об оригинальном лице идентифицируемого человека.

Наиболее распространённым и сложным видом атак является атака через электронные носители, пример которых представлен на рисунке 1.1. Это связано с тем, что, во-первых, данный вид атак не предполагает длительной подготовки (достаточно иметь видеофрагмент или фотографию идентифицируемого человека) и в случае с видеофрагментом отпадает важный признак статичности мимики лица, по которому довольно легко обнаружить физические носители.



Рисунок 1.1 – Пример настоящих лиц и фотографий экранов мобильных устройств с лицом человека.

По этой причине в данной работе рассматривались атаки с использованием мобильных устройств.

Метрикой оценки качества алгоритмов обнаружения подлинных лиц является Equal Error Rate (Равная степень ошибок). Данная метрика определяется как точка пересечения кривых FAR () и FRR (), что представлено на рисунке 1.2. Метрика (Степень ложных допусков) эквивалентна метрике

(1.1), тогда её кривая эквивалентна кривой, построенной также относительно метрики (Чувствительность) или также называемой (1.3) .Метрика (Степень ложных отказов) эквивалентна метрике (1.2) и строится аналогичным с кривой образом. Тогда, точка пересечения этих двух кривых будет соответствовать порогу классификации, при котором достигается минимально возможные одновременные значения данных метрик.

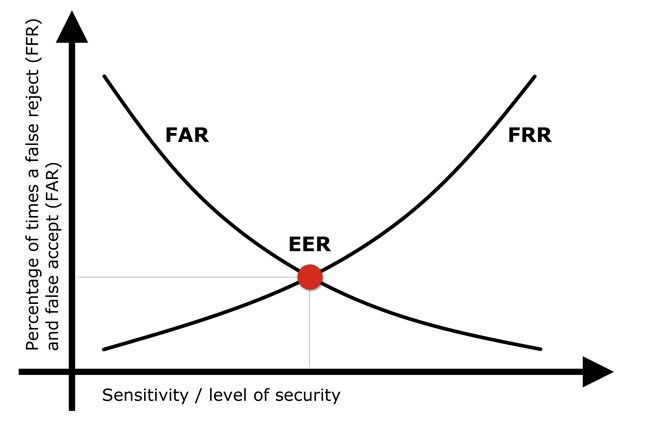


Рисунок 1.2. – Кривые метрик FAR и FRR.

где – ложноположительные результаты, кол-во атак, неверно принятых за реальных людей;

– истинно отрицательные результаты, кол-во атак верно принятые за атаки;

где – ложноотрицательные результаты, кол-во людей, неверно принятых за атаки;

– истинно положительные результаты, кол-во людей, верно принятых за людей;

где – ложноположительные результаты, кол-во атак, неверно принятых за реальных людей;

– истинно отрицательные результаты, кол-во атак верно принятые за атаки;

# 2 Набор данных

В данной работе использовался набор данных *LCC FASD,* представленный исследователями в работе [1]. Отличительной особенностью набора данных является наличие атак, осуществленных с различных моделей электронных устройств, и наличие различных моделей мобильных устройств для осуществления аутентификации через камеру, рисунки 2.1 и 2.2.

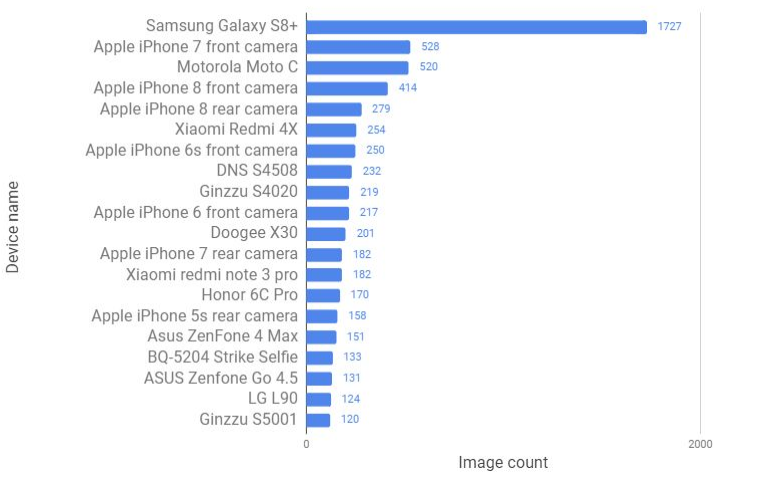


Рисунок 2.1 – Диаграмма частот топ-20 мобильных устройств, с которых осуществлялась аутентификация.

Набор данных содержит более 18 тысяч изображений, из которых более 16 тысяч – атаки. Так как данная работа концентрируется на исследовании признаков изображений, а не на обучении максимально робастного и точного алгоритма, для работы был создан набор данных, содержащий случайно выбранные 200 изображений набора данных *LCC FASD,* из которых 100 изображений – реальные лица, 100 – атаки на систему аутентификации.

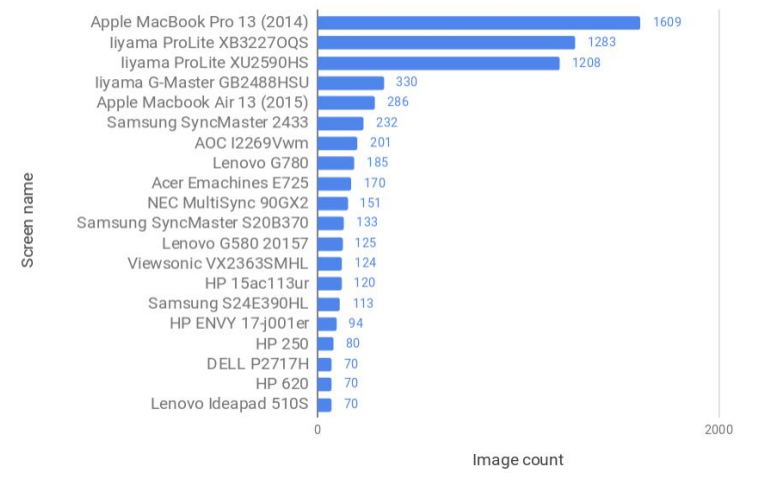


Рисунок 2.2 – Диаграмма частот топ-20 электронных устройств, с которых осуществлялась атака.

# 3 Синтез признаков

Синтез признаков для обнаружения атак с использованием экранов мобильных устройств происходит на основе артефактов видео захвата экранов. Одним из основных артефактов является муар. Муаром называется результат интерференции двух или более периодических структур, имеющих различные пространственные частоты. Для человека этот дефект воспринимается как полосатость изображения.

Также фотографиям экранов характерны узоры текстуры. Формирование этих текстур происходит из-за совокупности регулярных крошечных структур, появляющихся на поверхности экранов, уникальной схемы возбуждения с обратной полярностью и периодической перезарядки конденсаторов КМОП, которые влияют на яркость крошечных ЖК-ячеек.

Помимо этого, для повторно сфотографированных через экран мониторов или мобильных устройств изображений характерны цветовые артефакты, поскольку цветопередача современных экранов часто ограничена частью видимого спектра.

В статье [2] на основе данных артефактов были предложены следующие признаки:

- Статистики (выборочное среднее (3.1) и стандартное отклонение (3.2)) Вейвлет-преобразования. Двухмерное Вейвлет-преобразование изображения (рисунок 3.1) возвращает четыре компоненты изображения: низкочастотную, высокочастотную по горизонтали, по вертикали и по диагонали. Муар артефакты характеризуются наличием высокочастотных артефактов, поэтому низкочастотная компонента Вейвлет-преобразования не используется для синтеза признаков. Также, муар артефакты часто имеют горизонтальную направленность, но из-за того, что изображение может быть получено под некоторым углом относительно экрана, остальные компоненты Вейвлет-преобразования также используются для сбора статистик. Выбор статистик объясняется тем, что наличие высокочастотных компонент в изображении напрямую влияет как на выборочное среднее компоненты (оно будет смещаться вправо), так и на стандартное отклонение (оно будет увеличиваться).

(3.1)

где – выборочное среднее,

– размер выборки,

- – сумма всех значений выборки.

(3.2)

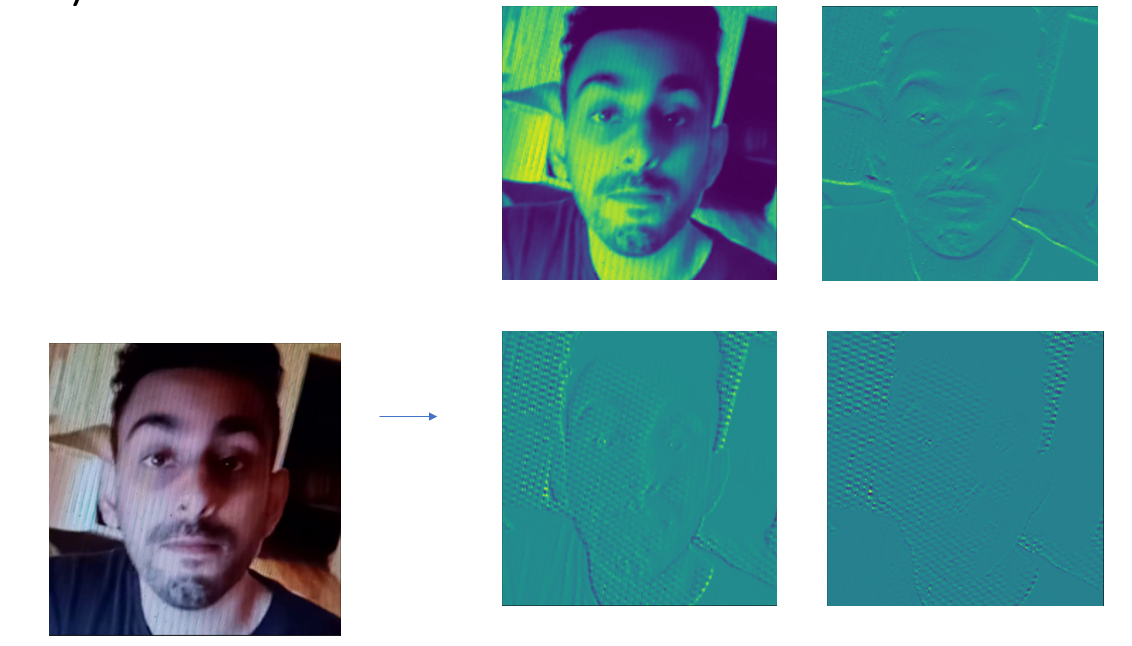


Рисунок 3.1 – Пример Вейвлет-преобразования изображения (преобразование применялось к черно-белому изображению с последующей колоризацией результатов преобразования).

- Гистограмма LBP (Local Binary Pattern) дескриптора. С помощью дескриптора LBP вычисляется нормализованная гистограмма однородных паттернов изображения. Для этого задают три параметра дескриптора: радиус (*R*), кол-во точек (*P*), рисунок 3.2, количество интервалов гистограммы. Для каждой точки изображения вычисляют разницу между рассматриваемой точкой и точкой, лежащей на расстоянии радиуса от рассматриваемой, если разность больше нуля, присваиваем точке значение 1, если меньше – 0. «Вытягиваем» значения точек на радиусе в двоичное число, преобразуем его в десятеричное, и заносим в гистограмму нашего признака, далее, преобразуем гистограмму к заданному значению интервалов.

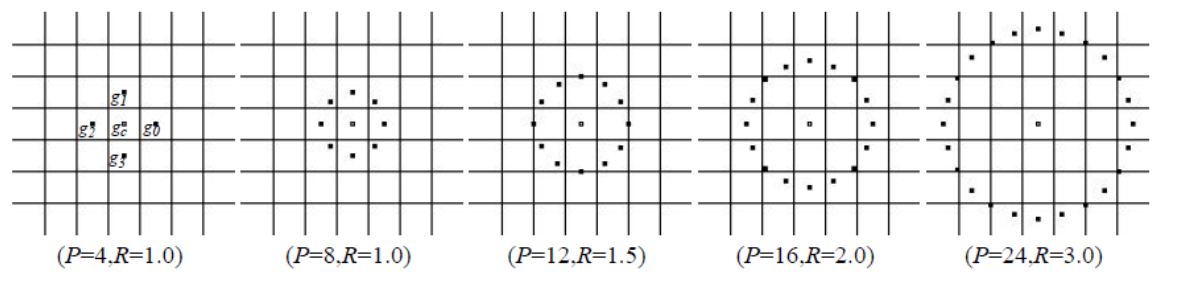


Рисунок 3.2 – LBP дескриптор в зависимости от значений параметров кол-ва точек и радуиса

В работе [3], где был впервые представлен данный дескриптор, были проведены эксперименты с различными векторами параметров дескриптора. Оценка производилась для задачи классификации изображений по метрике точности. По результатам исследования наилучшие результаты продемонстрировал вектор признаков, основанный на двух гистограммах LBP, взятых с параметрами (*P, R, nBins*): *(8, 1, 10), (8, 1, 26), (24, 3, 10), (24, 3, 26)*. В данной работе также были взяты признаки с перечисленными наборами параметров. Пример гистограмм LBP с параметрами *(8, 1, 10)* для настоящего лица и атаки представлен на рисунке 3.3.

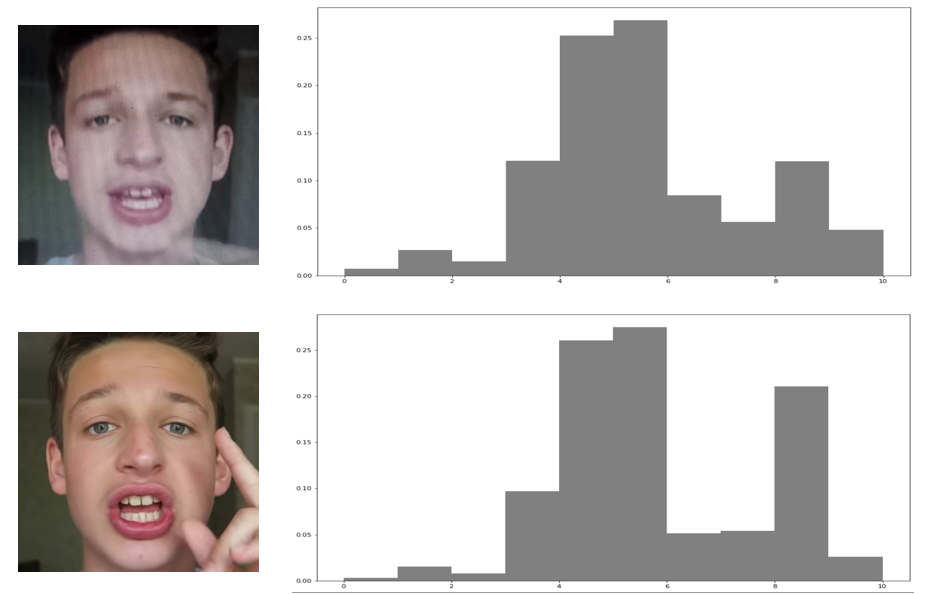


Рисунок 3.3 – Гистограммы LBP с параметрами *(8, 1, 10)* для настоящего лица (снизу) и атаки (сверху).

- Цветовые признаки. Для синтеза цветовых признаков было решено использовать вышеперечисленные дескрипторы к различным цветовым пространствам изображения: RGB, HSV, YCbCr. Также дескриптор LBP применялся к одноканальному черно-белому изображению.

Тогда общая длина вектора признаков составит:

признака

# 4 Отбор статистически значимых признаков

Так как вектор признаков имеет достаточно большую длину, имеет смысл провести тест на статистическую значимость признаков, который позволит сократить длину вектора признака без значимой потери информации для алгоритмов.

Статистический тест проверяет нулевую гипотезу о том, что выборки признака, разделенные в зависимости от своего класса (атака/настоящее лицо) имеют равные средние в генеральной совокупности, то есть принадлежат одному распределению. Соответственно, альтернативная гипотеза утверждает обратное, что, при подтверждении альтернативной гипотезы с заданным уровнем значимости, равным 0,05, даст полагать данный признак статистически значимым для последующей работы алгоритмов.

Анализ на статистическую значимость в данной работе проводится двумя методами: однофакторный дисперсионный анализ ANOVA и Bootstrap.

Перед проведением проверок на выявление статистически значимых признаков, были рассчитаны коэффициенты корреляции Пирсона для всех пар признаков, в результате чего было выявлено, что гистограммы LBP с наборами параметров *(8, 1, 10)* и *(8, 1, 26)* идентичны, было решено удалить вектор параметров *(8, 1, 10)* из набора признаков. Таким образом, длина вектора признаков стала 674.

## 4.1 Однофакторный дисперсионный анализ ANOVA

Идея метода заключается в том, что изменчивость имеющихся данных () может быть обусловлена изменчивостью внутри групп () и изменчивостью между группами ():

* Если большая часть общей изменчивости () обеспечивается благодаря изменчивости между группами (), это означает, что группы различаются между собой.
* Если большая часть общей изменчивости () обеспечивается благодаря внутригрупповой изменчивости (), то это говорит о том, что группы не различаются.
* Если межгрупповой показатель изменчивости () значительно превышает внтуригрупповой (), то весьма вероятно что как минимум два средних значения отличаются друг от друга.

, (4.1)

где – значение -го элемента -ой группы, – среднее по всем группам.

, (4.2)

где – выборочное среднее -ой группы, – среднее по всем группам.

, (4.3)

где – значение -го элемента -ой группы, – выборочное среднее -ой группы.

Усредненное значение межгрупповой суммы квадратов называется межгрупповым средним квадратом (межгрупповая дисперсия):

где – число степеней свободы, равное:

Внутригрупповой средний квадрат (внутригрупповая дисперсия):

где – число степеней свободы, равное:

Далее вычисляется *F*-критерий Фишера:

Зная критерий Фишера, межгрупповую и внутригрупповую степени свободы можно найти *p*-уровень значимости для *F-*распределения. Значение *p*-уровня значимости меньше 0,05 будут соответствовать маловероятному, то есть статистически не значимому, сходству двух групп.

Данный метод имеет ряд ограничений, а именно:

- Выборки должны быть независимы.

- Каждая из выборок принадлежит нормальному распределению.

- Гомоскедастичность.

Проверку на нормальность выборок осуществляем обобщенным критерием Д’Агостино-Пирсона, согласно результатам этого теста, все выборки каждого из признаков принадлежат нормальному распределению. Проверка на гомоскедастичность осуществлялась критерием Барлетта, согласно проверке, все признаки оказались гомоскедастичны.

Код проверки выборок на нормальность и гомоскедастичность, а также расчет *p*-уровня значимости каждого признака приложен в приложении А.

По результатам расчета *p*-уровня значимости были отфильтрованы признаки, *p*-уровень значимости которых больше 0,05. В результате, длина вектора признаков стала равна 265.

## 4.2 Bootstrap

Идея метода bootstrap заключается в том, чтобы случайным образом многократно формировать выборки из наших групп признаков методом Монте-Карло, собирать из данных bootstrap какую-то статистику и находить её доверительные интервалы. В нашем случае было решено сформулировать нулевую гипотезу как равенство медиан двух выборок одного признака, разделенных по классам.

Для этого 1000 раз методом Монте-Карло были сформированы случайные выборки размером, равным исходным выборкам, и на каждой итерации рассчитывалась разница медиан двух выборок. Далее, для полученных значений разности медиан находился доверительный интервал по *t-*распределению Стъюдента:

(4.9)

где –коэффициент Стьюдента, зависящий от числа измерений n и доверительной вероятности a. Примем доверительную вероятность равную a=0.95

Тогда, статистически значимыми признаками будут являться те признаки, разность медиан выборок которых имеет доверительный интервал, не включающий в себя ноль. Код проверки на статистическую значимость признаков методом bootstrap приведен в приложении Б. По результатам теста длина вектора признаков сократилась до 255.

# 5 Анализ значимости признаков

После проведения тестов на стат. значимость было решено сформировать четыре набора данных. Первый набор данных включает в себя все синтезированные признаки, за исключением вектора признаков дескриптора LBP с параметрами *(8, 1, 10)* (674 признака), второй – признаки, прошедшие однофакторный дисперсионный анализ ANOVA (265 признаков), третий – признаки, прошедшие bootstrap (255 признаков), четвертый – признаки, прошедшие и ANOVA, и bootstrap (214 признаков).

Для каждого набора данных вычислялись, во-первых, вектор значимости признаков для алгоритма, и, во-вторых, значение метрики *EER.* Далее, была сформирована таблица, где каждому признаку соответствовал определенный уровень значимости в алгоритме. Для определения уровня значимости признака в алгоритме были выбраны два алгоритма классификации: Случайный лес и Логистическая регрессия, алгоритм Логистической регрессии также использовался для отбора признаков, путём включения в него *l1*-регуляризации.

Идея алгоритма Случайный лес заключается в ассемблировании Деревьев принятий решений, суть которых в нахождении предикатов, на основе которых можно разделить выборку таким образом, чтобы среднее значение энтропии уменьшалось. Энтропия рассчитывается по формуле 5.1:

где – вероятность *i*-го класса в рассматриваемом узле.

Таким образом, на первом этапе рассчитывается энтропия всей выборки, далее итеративно перебираются все признаки и все пороговые значения этих признаков, на каждой итерации рассчитывается энтропия каждой из двух получившихся после разбиения выборок, и рассчитывается *Information Gain* (5.2), то есть значение уменьшения энтропии после разбиения выборки. После завершения цикла выбирается тот признак и его пороговое значение, которые дают максимальное уменьшение выборки. Это значение можно также интерпретировать как значимость признака для алгоритма.

где – энтропия выборки до разбиения, – взвешенное по кол-ву объектов в выборке среднее энтропий двух получившихся после разбиения выборок.

Случайны лес является ансамблем деревьев решений. Для каждого из *N* деревьев, где *N –* настраиваемый параметр кол-ва деревьев решений, методом bootstrap берется выборка объектов из набора данных, случайным образом перед каждым разбиением выбирается признаков, где – длина вектора признаков. Далее, каждое дерево независимо обучается на своих выборках, предсказанный класс в Случайном лесу — это класс с самой высокой оценкой средней вероятности по деревьям.

Логистическая регрессия – обобщённая линейная модель (Generalized Linear Models) классификации. Данная модель, как следует из названия, строится на основе линейной модели (5.3), но в классическом своём виде линейная модель служит для предсказания нормально распределенной величины, то есть интервал возможных значений предсказываемой величины – .

Чтобы перейти к предсказанию вероятностей, во-первых, модель модифицируют так, чтобы предсказывать не нормально распределенную величину, а биноминальную величину, вероятность которой определяется, как шанс (5.4).

Но интервал значений шанса - , поэтому модель предсказывает не сами шансы, а их логиты биноминальных вероятностей (5.5), что ограничивает область предсказываемых значений на интервале .

Тогда коэффициенты этой обобщённой линейной модели можно интерпретировать как вклад соответствующего признака в увеличение или уменьшение шанса события. В данной работе исследуется только сила значимости признаков, поэтому знак весового коэффициента не учитывается.

Логистическая регрессия в данной работе также выступает в роли фильтра признаков, этому способствует внедрение *l1-* регуляризации (5.6), которая имеет тенденцию занулять признаки во время оптимизации. По результатам обучения алгоритма, было получено 435 признаков.

Поиск оптимальных параметров алгоритмов проводился по сетке (GridSearch) по минимизации метрики *EER*. Код обучения алгоритмов предоставлен в приложении В.

По результатам работы были составлены таблицы значимости признаков в зависимости от их дескрипторов и параметров дескрипторов (таблицы 2-5) и таблица качества работы алгоритмов по метрике *EER* (таблица 1). В таблицах принято следующее обозначение:

- Случайный лес, обученный на всех признаках, - СЛ 1;

- Случайный лес, обученный на признаках, отобранных и по ANOVA и по bootstrap, - СЛ 2;

- Случайный лес, обученный на признаках, отобранных только по ANOVA, - СЛ 3;

- Случайный лес, обученный на признаках, отобранных только по bootstrap, - СЛ4.

Таблица 1 – Качество работы алгоритмов по метрике *EER*.

|  |  |
| --- | --- |
| Название алгоритма | *EER, %* |
| СЛ 1 | 10,2 |
| СЛ 2 | 10,3 |
| СЛ 3 | 13,8 |
| СЛ 4 | 12,9 |
| Логистическая регрессия | 6,9 |

По данным из таблицы 1 можно сделать вывод, что набор признаков, отобранный по обоим статистическим тестам показ результаты, близкие к алгоритму, обученному на полном векторе признаков. Тем не менее, логистическая регрессия продемонстрировала значимо более высокие результаты. Следует отметить, что в данной работе участвовала выборка из двухсот изображений, поэтому нельзя с уверенностью сказать, являются ли данные результаты репрезентативны.

Таблица 2 – Значимость признаков в зависимости от дескриптора.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Дескриптор** | **Значимость в СЛ 1** | **Значимость в СЛ 2** | **Значимость в СЛ 3** | **Значимость в СЛ 4** | **Значимость в лог. регрессии** |
| **LBP** | 0,00133 | 0,001242 | 0,001282 | 0,001254 | 0,034473 |
| **WS** | 0,003243 | 0,004263 | 0,003801 | 0,004115 | 0,052401 |

Результаты таблицы 2 демонстрируют более высокую значимость для признаков дескриптора Вейвлет-преобразований. Это объясняется в первую очередь не значимостью самих признаков, а кол-вом незначимых признаков у LBP дескриптора, так как LBP признаки представляют из себя гистограммы, в которых значимость имеют малая доля интервалов.

Таблица 3 – Значимость признаков в зависимости от цветовой модели изображения.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Цветовая модель** | **Значимость в СЛ 1** | **Значимость в СЛ 2** | **Значимость в СЛ 3** | **Значимость в СЛ 4** | **Значимость в лог. регрессии** |
| **BGR** | 0,001062 | 0,001017 | 0,000929 | 0,00105 | 0,027848 |
| **GRAY** | 0,000954 | 0,000845 | 0,000827 | 0,000965 | 0,031512 |
| **HSV** | 0,001784 | 0,001725 | 0,001811 | 0,001744 | 0,041560 |
| **YCrCb** | 0,001766 | 0,001904 | 0,00191 | 0,001816 | 0,039656 |

По результатам таблицы 3 можно сделать вывод о более высокой значимости у признаков, собранных с HSV и YCrCb изображений. Чтобы понять причину более высокой значимости данных признаков, обратимся к таблице 4. Из данной таблицы можно более конкретно определить, что самую высокую значимость имеют признаки из канала *Saturation* (Насыщенность) HSV и канала *Cb* (Синяя цветоразностная компонента) YCrCb. Это можно объяснить тем, что ограниченная цветопередача экранов смартфонов влияет на насыщенность цветов, также при фотографировании экранов характерны оттенки синего, которые лучше регистрируются в цветоразностной компоненте, нежели чем в синем канале *BGR* изображения.

Таблица 4 – Значимость признаков в зависимости от канал соответствующей цветовой модели.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Цветовая модель** | **Канал** | **Значимость в СЛ 1** | **Значимость в СЛ 2** | **Значимость в СЛ 3** | **Значимость в СЛ 4** | **Значимость в лог. регрессии** |
| **BGR** | **CH0** | 0,001029 | 0,000919 | 0,000881 | 0,000841 | 0,024963 |
| **CH1** | 0,00115 | 0,001185 | 0,001033 | 0,00131 | 0,032577 |
| **CH2** | 0,001008 | 0,000946 | 0,000874 | 0,000998 | 0,026005 |
| **GRAY** | **CH0** | 0,000954 | 0,000845 | 0,000827 | 0,000965 | 0,031512 |
| **HSV** | **CH0** | 0,001559 | 0,001249 | 0,001261 | 0,001132 | 0,036136 |
| **CH1** | 0,00249 | 0,003055 | 0,003187 | 0,003058 | 0,061612 |
| **CH2** | 0,001302 | 0,00087 | 0,000985 | 0,001041 | 0,026932 |
| **YCrCb** | **CH0** | 0,00153 | 0,001116 | 0,001205 | 0,001019 | 0,028362 |
| **CH1** | 0,001366 | 0,001486 | 0,001501 | 0,001517 | 0,028814 |
| **CH2** | 0,002402 | 0,003111 | 0,003024 | 0,00291 | 0,061792 |

По таблице 5 можно сделать выводы, что, во-первых, более детальный фильтр LBP с 24 точками и радиусом 3 предоставляет алгоритму больше полезной информации. Во-вторых, нельзя сделать вывод относительного того, какая высокочастотная компонента Вейвлет-преобразования однозначно более информативна для алгоритмов. Например, для Логистической регрессии вертикальная высокочастотная компонента оказалась значительно более информативнее, чем остальные, а для Случайного леса с признаками, прошедшими и ANOVA, и bootstrap, более информативной оказалась горизонтальная высокочастотная компонента.

Таблица 5 – Значимость признаков в зависимости от атрибутов соответствующих дескрипторов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Дескриптор** | **Атрибуты дескриптора** | **Значимость в СЛ 1** | **Значимость в СЛ 2** | **Значимость в СЛ 3** | **Значимость в СЛ 4** | **Значимость в лог. регрессии** |
| **LBP** | **SC24+3** | 0,001923 | 0,001817 | 0,00188 | 0,001852 | 0,048969 |
| **SC8+1** | 0,00051 | 0,000445 | 0,000453 | 0,000427 | 0,014402 |
| **WS** | **HH** | 0,00376 | 0,003966 | 0,003599 | 0,004019 | 0,033344 |
| **HL** | 0,002687 | 0,003768 | 0,003193 | 0,004245 | 0,076793 |
| **LH** | 0,003283 | 0,005055 | 0,00461 | 0,004083 | 0,047066 |

По таблице 6 можно заметить, что более длинные интервалы гистограммы оказались информативнее гистограмм с 26-ю интервалами. То есть обобщение информации уже на уровне формирования вектора признака для LBP дескриптора способствует увеличению информативности признака. Для Вейвлет-преобразований всё по-прежнему не однозначно, для Случайных лесов более информативны однозначно стандартные отклонения высокочастотных компонент, причем диагональные и горизонтальные компоненты информативнее вертикальных. Для Логистической регрессии, наоборот, более информативной оказалась вертикальная компонента, причем статистика выборочного среднего, а не стандартное отклонение, хотя для остальных компонент также характерна бОльшая информативность для стандартных отклонений.

Таблица 6 – Значимость признаков в зависимости от дополнительных атрибутов соответствующих дескрипторов.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Дескриптор** | **Атрибуты дескр.** | **Доп. атрибуты дескр.** | **Значимость в СЛ 1** | **Значимость в СЛ 2** | **Значимость в СЛ 3** | **Значимость в СЛ 4** | **Значимость в лог. регрессии** |
| **LBP** | **SC24+3** | **COUNTBINS10** | 0,003843 | 0,003943 | 0,004096 | 0,004218 | 0,069065 |
| **COUNTBINS26** | 0,001185 | 0,000999 | 0,001028 | 0,000942 | 0,041239 |
| **SC8+1** | **COUNTBINS26** | 0,00051 | 0,000445 | 0,000453 | 0,000427 | 0,014402 |
| **WS** | **HH** | **mean** | 0,001726 | 0,002048 | 0,001264 | 0,001691 | 0,025102 |
| **std** | 0,005794 | 0,005883 | 0,005935 | 0,006347 | 0,041586 |
| **HL** | **mean** | 0,00179 | 0,002767 | 0,001933 | 0,002663 | 0,081367 |
| **std** | 0,003584 | 0,004769 | 0,004453 | 0,005826 | 0,072219 |
| **LH** | **mean** | 0,002444 | 0,003085 | 0,002266 | 0,002972 | 0,024787 |
| **std** | 0,004121 | 0,007025 | 0,006954 | 0,005193 | 0,069345 |

# ЗАКЛЮЧЕНИЕ

В данной работе была сформулирована задача синтезирования признаков и определения их значимости в алгоритмах машинного обучения для задачи обнаружения атак в системах видео аутентификации. Была сформулирована задача, проведен обзор набора данных, дескрипторов, на основе которых синтезированы признаки. Были проведены статистические тесты для определения статистической значимости полученных признаков. Также было проведено обучение алгоритмов Случайный лес и Логистическая регрессия на различных наборах векторов признаков.

По результатам анализа значимости признаков можно сделать вывод, что цветовые модели *HSV* и *YCrCb* в среднем более информативны, чем *BGR.* Также, увеличение параметров кол-ва точек и радиуса в дескрипторе LBP способствует увеличению информативности получаемых данных, но на уровне гистограммы, наоборот, увеличение кол-ва интервалов, то есть сохранение информации, способствует уменьшению информативности. Для Вейвлет-преобразований не было получено однозначных выводов, но можно сделать вывод, что стандартные отклонения диагональных и горизонтальных компонент более информативны в алгоритмах Случайных лесов, а в алгоритме Логистической регрессии, наоборот, более информативно выборочное среднее вертикальной компоненты.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. D. Timoshenko, K. Simonchik Large Crowdcollected Facial Anti-Spoofing Dataset// CSIT Conference 2019, Yerevan, Armenia, September 23-27

2. Hong Cao, Alex C. Kot IDENTIFICATION OF RECAPTURED PHOTOGRAPHS ON LCD SCREENS // ICASSP 2010: IEEE International Conference on Acoustics, Speech and Signal Processing

3. T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Textureclassification with Local Binary Patterns," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-987, 2002

# Приложение А

Код на языке Python проверки выборок на принадлежность нормальному распределению и гомоскедастичность, и расчёт *p*-уровня значимости для каждого признака.

from scipy import stats

for feat in X.columns:

unnormal\_feats = []

unhomosced\_feats = []

feat\_df = pd.concat([X[feat], y], axis=1)

sample\_1 = feat\_df[feat\_df.label == 1][feat]

sample\_2 = feat\_df[feat\_df.label == 0][feat]

stat\_1, p\_value\_1 = stats.normaltest(sample\_1)

stat\_2, p\_value\_2 = stats.normaltest(sample\_2)

if p\_value\_1 < 0.05 or p\_value\_2 < 0.05:

unnormal\_feats.append(feat)

continue

hstat, hp\_value = stats.bartlett(sample\_1, sample\_2)

if hp\_value < 0.05:

unhomosced\_feats.append(feat)

from sklearn.feature\_selection import f\_classif

f\_score, f\_pval = f\_classif(X, y)

f\_scores\_data = pd.DataFrame(np.concatenate([np.expand\_dims(f\_score, -1), np.expand\_dims(f\_pval, -1)], axis=1), columns=['f\_score', 'pval'], index=X.columns)

# Приложение Б

Код на языке Python проверки на статистическую значимость признаков методом bootstrap.

import bootstrapped.bootstrap as bs

import bootstrapped.compare\_functions as bs\_compare

import bootstrapped.stats\_functions as bs\_stats

stat\_significant = [] for feat in X.columns: feat\_df = pd.concat([X[feat], y], axis=1) sample\_1 = feat\_df[feat\_df.label == 1][feat].to\_numpy()

sample\_2 = feat\_df[feat\_df.label == 0][feat].to\_numpy()

conf\_interval = bs.bootstrap\_ab( sample\_1, sample\_2, stat\_func=bs\_stats.median, compare\_func=bs\_compare.difference, iteration\_batch\_size=None,) lower\_bounder = conf\_interval.lower\_bound

upper\_bounder = conf\_interval.upper\_bound

if (lower\_bounder <= 0) & (upper\_bounder>=0):

stat\_significant.append(0)

else:

stat\_significant.append(1)

# Приложение В

Поиск оптимальных параметров и обучение алгоритмов Случайный лес и Логистической регрессии

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split, GridSearchCV, RandomizedSearchCV

from sklearn.metrics import make\_scorer

eer\_score = make\_scorer(eer, greater\_is\_better=False)

def random\_trees\_feats\_range(X\_data):

X\_filtered = X\_data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_filtered, y, test\_size=.3, random\_state=42)

n\_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]

max\_features = ['auto', 'sqrt'] max\_depth = [int(x) for x in np.linspace(10, 110, num = 11)]

max\_depth.append(None)

min\_samples\_split = [2, 5, 10]

min\_samples\_leaf = [1, 2, 4]

bootstrap = [True, False]

random\_grid = {'n\_estimators': n\_estimators,

'max\_features': max\_features,

'max\_depth': max\_depth,

'min\_samples\_split': min\_samples\_split, 'min\_samples\_leaf': min\_samples\_leaf,

'bootstrap': bootstrap}

clf = RandomForestClassifier(criterion='entropy', random\_state=42) clf\_random = RandomizedSearchCV(estimator = clf, param\_distributions = random\_grid, n\_iter = 100, cv = 5, verbose=2, random\_state=42, n\_jobs = -1)

clf\_random.fit(X\_train, y\_train)

best\_tree = clf\_random.best\_estimator\_

y\_pred = best\_tree.predict\_proba(X\_test)

print('EER:', eer(y\_test, y\_pred[:,1]))

print('Best params:', best\_tree.get\_params)

range\_feat\_df = pd.DataFrame(best\_tree.feature\_importances\_, index=X\_filtered.columns, columns=['tree\_feat\_score']). \ sort\_values(by='tree\_feat\_score', ascending=False) return best\_tree, range\_feat\_df

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

import warnings warnings.filterwarnings('ignore')

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=.3, random\_state=42)

stand\_scaler = StandardScaler()

stand\_scaler.fit(X\_train)

X\_stand\_train = stand\_scaler.transform(X\_train)

X\_stand\_test = stand\_scaler.transform(X\_test)

log\_reg = LogisticRegression(penalty='l1', random\_state=0)

params = {'solver':['liblinear', 'saga'], 'C': np.arange(0.1, 10.0, 0.5)}

clf = GridSearchCV(log\_reg, param\_grid=params, cv=5, scoring=eer\_score) clf.fit(X\_stand\_train, y\_train)

log\_reg = clf.best\_estimator\_ y\_pred = log\_reg.predict\_proba(X\_stand\_test) print('EER:', eer(y\_test, y\_pred[:,1]))